


Web Service {SERVER}

- Persiapan database dan tabel
- Persiapan aplikasi (folder project dan instalasi Codeiginter, RESTFull)
 - API

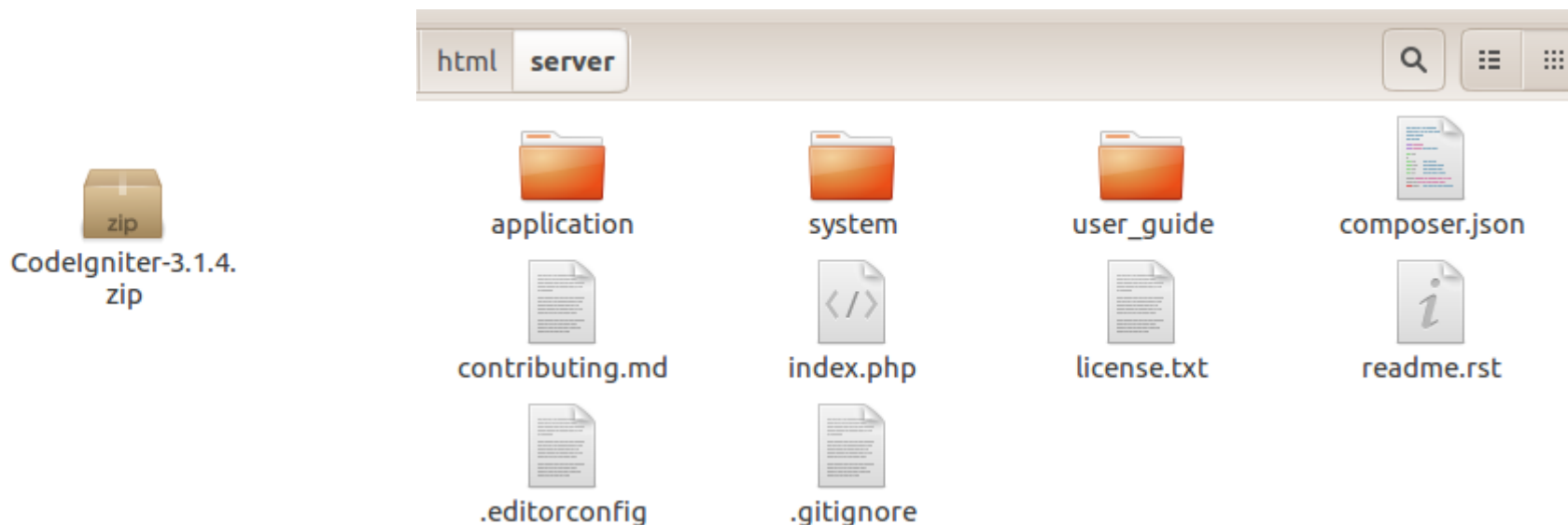
Database & Tabel

- Buat database “**webservice**” dan tabel “**customer**” dengan struktur berikut:

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	customerID 	int(11)			No	None	AUTO_INCREMENT
2	customerName	varchar(40)	latin1_swedish_ci		No	None	
3	customerCity	text	latin1_swedish_ci		No	None	
4	customerPhone	varchar(12)	latin1_swedish_ci		No	None	

Persiapan file project SERVER

- Download Codeigniter (CI) terbaru
 - <https://www.codeigniter.com/download>
 - <https://github.com/bernardvery/webservice>
- Buat folder dengan nama “**server**” pada direktori web server (misalkan: d:\xampp\htdocs\server) dan Extract CI ke folder tersebut



Library REST Server

- Download file library restserver di:
 - <https://github.com/chriskacerguis/codeigniter-restserver>
 - <https://github.com/bernardvery/webservice>
- Extract lalu copy kan file berikut:

codeigniter-restserver-master/application/config/**rest.php**

---- copy ke -----

server/application/config/**rest.php**

codeigniter-restserver-master/application/libraries/**Format.php**

---- copy ke -----

server/application/libraries/**Format.php**

codeigniter-restserver-master/application/libraries/**REST_Controller.php**

---- copy ke -----

server/application/libraries/**REST_Controller.php**

codeigniter-restserver-master/application/language/english/**rest_controller_lang.php**

---- copy ke -----

server/application/language/english/**rest_controller_lang.php**



codeigniter-
restserver-master.
zip

Konfigurasi Database

- Modifikasi file “database.php” pada folder “application/config”

```
$db['default'] = array(
    'dsn'        => '',
    'hostname'   => 'localhost',
    'username'   => 'root',
    'password'   => '',
    'database'   => 'webservice',
    'dbdriver'   => 'mysqli',
    'dbprefix'   => '',
    'pconnect'   => FALSE,
    'db_debug'   => (ENVIRONMENT !== 'production'),
    'cache_on'   => FALSE,
```

Konfigurasi Autoload Library

- Modifikasi file “autoload.php” pada folder “application/config”, load library “database” secara default

```
44  /*
45  |  -----
46  |  Auto-load Libraries
47  |  -----
48  |  These are the classes located in system/libraries/ or your
49  |  application/libraries/ directory, with the addition of the
50  |  'database' library, which is somewhat of a special case.
51  |
52  |  Prototype:
53  |
54  |      $autoload['libraries'] = array('database', 'email', 'session');
55  |
56  |  You can also supply an alternative library name to be assigned
57  |  in the controller:
58  |
59  |      $autoload['libraries'] = array('user_agent' => 'ua');
60  */
61  $autoload['libraries'] = array('database');
62
```

API web service

- Buat controller dengan nama “Api.php” simpan pada folder “application/controllers”, buat class Api extends REST_Controller seperti berikut:

```
1 <?php
2 /**
3  * Webservice untuk data customer
4  * web service untuk melayani transaksi data customer.
5  */
6 require_once APPPATH . 'libraries/REST_Controller.php';
7 use Restserver\Libraries\REST_Controller;
8
9 class Api extends REST_Controller {
10     function __construct($config = 'rest') {
11         parent::__construct($config);
12     }
13
14 }
```

Request GET

- Tambahkan fungsi **customers_get()** pada class Api.php untuk melayani request **GET** dari client

```
function customers_get() {  
    $id = $this->get('id');  
    if ($id) {  
        $customer = $this->db->get_where('customer',  
            array('customerID' => $id))->result();  
    } else {  
        $customer = $this->db->get('customer')->result();  
    }  
    // generate response  
    if($customer){  
        $this->response($customer,200);  
    }else{  
        $this->response(array('status'=>'not found'),404);  
    }  
}
```


Request POST

- Tambahkan fungsi **customers_post()** pada class Api.php untuk melayani request **POST** dari client

```
function customers_post() {  
    $params = array(  
        'customerName' => $this->post('name'),  
        'customerPhone' => $this->post('phone'),  
        'customerCity' => $this->post('city'));  
    $process = $this->db->insert('customer', $params);  
    if($process){  
        // 201 artinya Successful creation of a resource.  
        $this->response(array('status'=>'succes'),201);  
    }else{  
        // 502 artinya Backend service failure (data store failure).  
        return $this->response(array('status'=>'fail'), 502);  
    }  
}
```

Request PUT

- Tambahkan fungsi **customers_put()** pada class Api.php untuk melayani request **PUT** dari client

```
function customers_put() {  
    $params = array(  
        'customerName' => $this->put('name'),  
        'customerPhone' => $this->put('phone'),  
        'customerCity' => $this->put('city'));  
    $this->db->where('customerID', $this->put('id'));  
    $execute = $this->db->update('customer', $params);  
    if($execute){  
        $this->response(array('status'=>'succes'),201);  
    }else{  
        return $this->response(array('status'=>'fail'), 502);  
    }  
}
```

Request DELETE

- Tambahkan fungsi **customers_delete()** pada class Api.php untuk melayani request **DELETE** dari client

```
function customers_delete() {  
    $this->db->where('customerID', $this->delete('id'));  
    $execute = $this->db->delete('customer');  
    if($execute){  
        $this->response(array('status'=>'succes'),201);  
    }else{  
        return $this->response(array('status'=>'fail'), 502);  
    }  
}
```

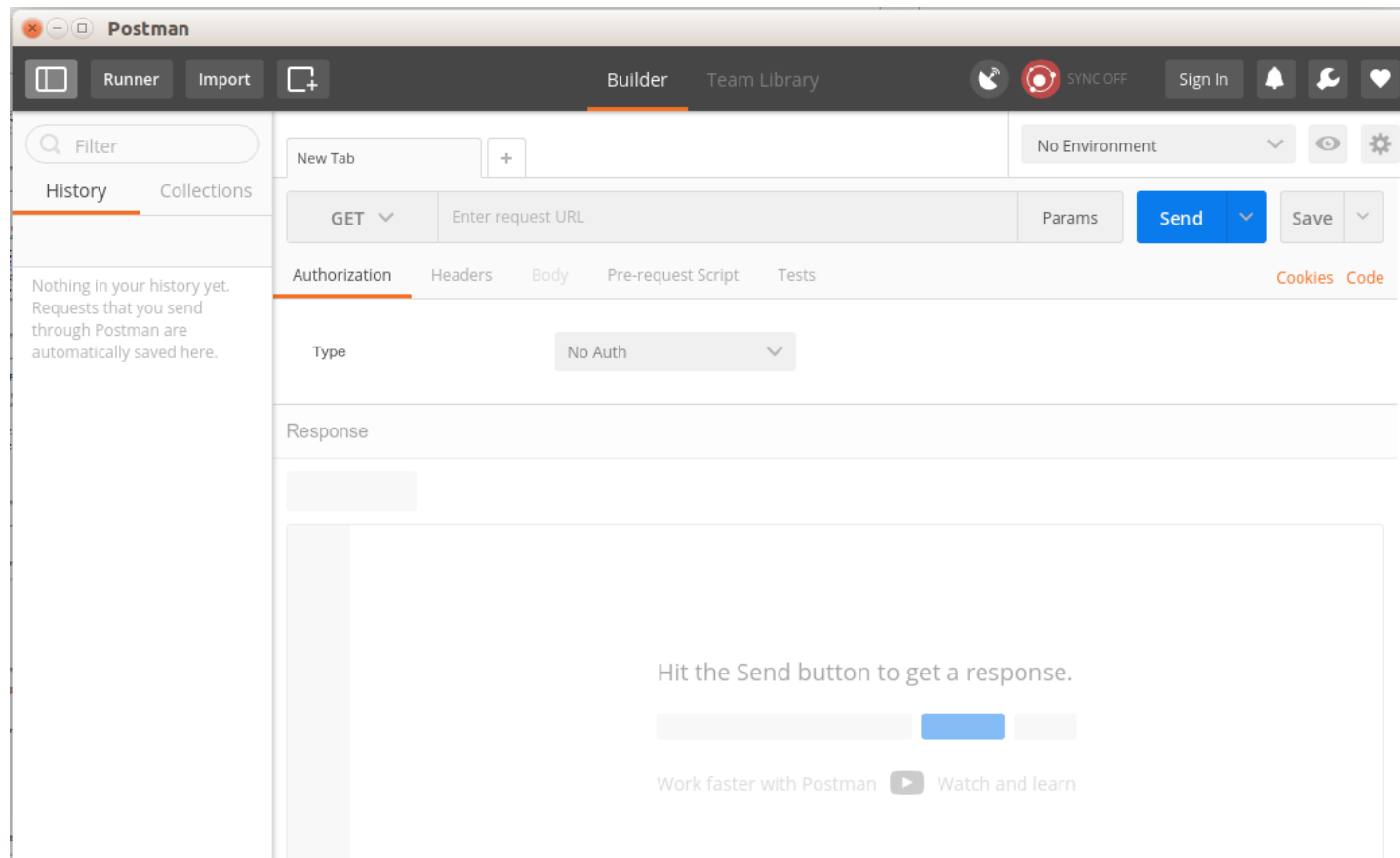
Uji Coba API (1)

- Isi tabel dengan data berikut:

NO	NAMA LENGKAP	ALAMAT	NO HP
1	Dian	semarang	0810000000
2	Eko	Jogja	0890000000
3	Rina	semarang	0850000000

Uji Coba API (2)

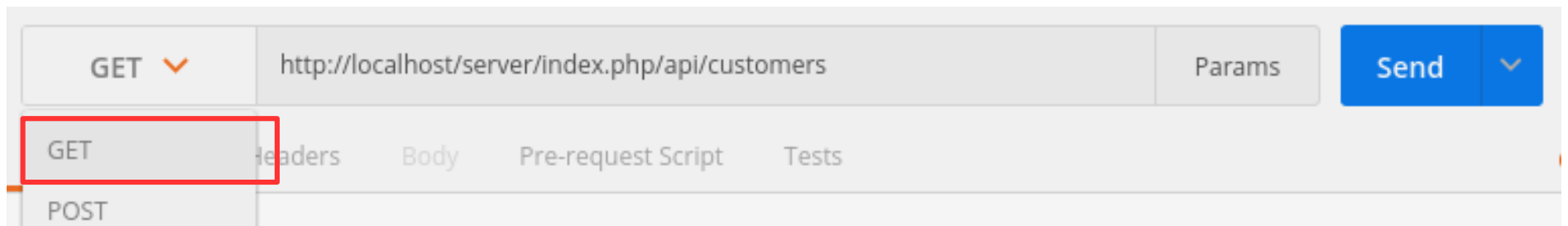
- Download aplikasi POSTMAN dan jalankan
 - <https://www.getpostman.com/>
 - <https://github.com/bernardvery/webservice>



Uji Coba API - GET (1)

Request → tampilkan semua record

- URL: <http://localhost/server/index.php/api/customers>
- Method: GET
- Klik tombol “Send”



Hasil Request

The screenshot displays the 'Body' tab of a web browser's developer tools. The response status is '200 OK', the time taken is '45 ms', and the size is '496 B'. The response body is a JSON array containing three customer records. The interface includes tabs for 'Body', 'Cookies', 'Headers (6)', and 'Tests'. Below these, there are options for 'Pretty', 'Raw', and 'Preview', with 'JSON' selected. A search icon is also present. The JSON data is as follows:

```
[
  {
    "customerID": "6",
    "customerName": "Dian",
    "customerCity": "semarang",
    "customerPhone": "0810000000"
  },
  {
    "customerID": "7",
    "customerName": "Eko",
    "customerCity": "Jogja",
    "customerPhone": "0890000000"
  },
  {
    "customerID": "8",
    "customerName": "Rina",
    "customerCity": "semarang",
    "customerPhone": "085000000"
  }
]
```

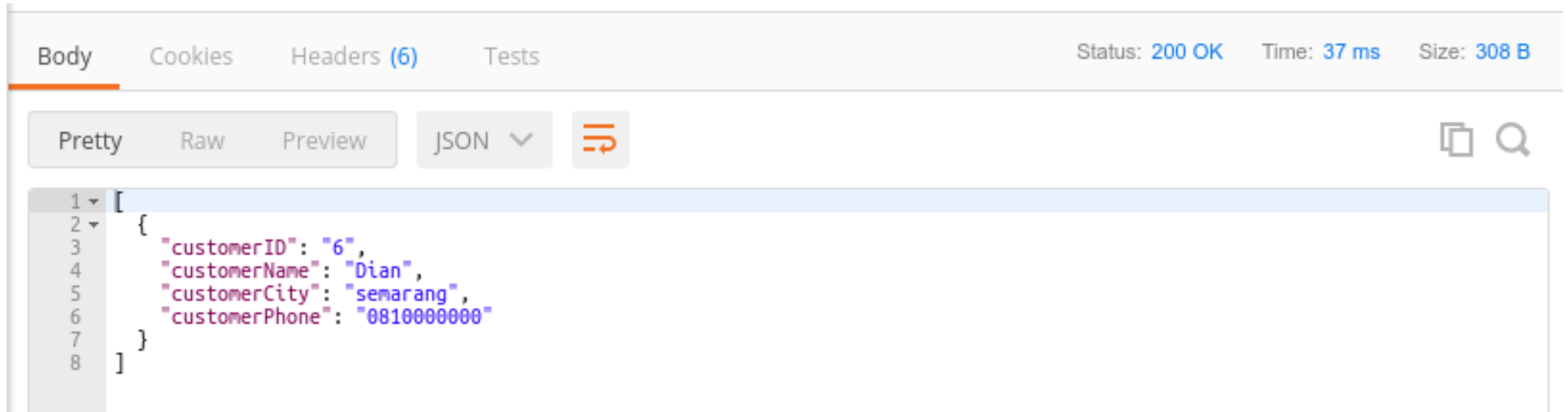
Uji Coba API - GET (2)

Request → tampilkan record tertentu

- URL: <http://localhost/server/index.php/api/customers>
- Method: GET
- Klik tombol “Params”, isikan parameter berikut: key=id, value=6
- Klik tombol “Send”

GET ▾	http://localhost/server/index.php/api/customers?id=6	Params	Send ▾
	Key	Value	
≡	id	6	

Hasil Request



The screenshot displays the 'Body' tab of a web browser's developer tools. The status bar at the top indicates a successful request with 'Status: 200 OK', a response time of 'Time: 37 ms', and a size of 'Size: 308 B'. The response body is shown in a 'Pretty' JSON format, containing a single object with the following fields: 'customerID' (6), 'customerName' (Dian), 'customerCity' (semarang), and 'customerPhone' (0810000000). The interface includes tabs for 'Body', 'Cookies', 'Headers (6)', and 'Tests', as well as buttons for 'Pretty', 'Raw', 'Preview', and a 'JSON' dropdown menu.

Body Cookies Headers (6) Tests Status: 200 OK Time: 37 ms Size: 308 B

Pretty Raw Preview JSON


```
[
  {
    "customerID": "6",
    "customerName": "Dian",
    "customerCity": "semarang",
    "customerPhone": "0810000000"
  }
]
```

Uji Coba format Output API (json | xml)

- URL: <http://localhost/server/index.php/api/customers>
- Method: GET
- Klik tombol “Params”, isikan parameter berikut:
key=format, value=json | xml *(isikan salah satu)*
- Klik tombol “Send”

GET ▾	http://localhost/server/index.php/api/customers?format=xml	Params	Send ▾
	Key	Value	
	format	xml	

Output format API

Pretty Raw Preview XML 

```
1 <?xml version="1.0" encoding="utf-8"?>|
2 <xml>
3   <item>
4     <customerID>6</customerID>
5     <customerName>Dian</customerName>
6     <customerCity>semarang</customerCity>
7     <customerPhone>08100000000</customerPhone>
8   </item>
9   <item>
10    <customerID>7</customerID>
11    <customerName>Eko</customerName>
12    <customerCity>Jogja</customerCity>
13    <customerPhone>08900000000</customerPhone>
14  </item>
15  <item>
16    <customerID>8</customerID>
17    <customerName>Rina</customerName>
18    <customerCity>semarang</customerCity>
19    <customerPhone>08500000000</customerPhone>
20  </item>
21 </xml>
```

Uji Coba API – POST (1)

Request → insert record baru

- URL: <http://localhost/server/index.php/api/customers>
- Method: POST
- Klik tab “Body”, pilih “form-data”, isikan parameter berikut:

Key	Value
name	Joni
city	Jakarta
phone	088000000000

- Klik tombol “Send”

Uji Coba API – POST (2)

The screenshot displays a REST client interface with the following components and annotations:

- 2**: Method dropdown menu set to **POST**.
- 1**: URL input field containing `http://localhost/server/index.php/api/customers`.
- 3**: Tab selection bar with **Body** selected.
- 4**: Content type selection area where **form-data** is selected.
- 5**: Table of request body data:

Key	Value
<input checked="" type="checkbox"/> name	Joni
<input checked="" type="checkbox"/> city	Jakarta
<input checked="" type="checkbox"/> phone	088000000000
New key	value

Below the table, the **Body** tab is active, showing the response in **JSON** format:

```
1 {  
2   "status": "succes"  
3 }
```

7 points to the response JSON, which is highlighted in blue.

6: **Send** button.

Status: **201 Created** Time: **122 ms**

Hasil request POST → input data ke server

Hasil POST

GET Params

Body Cookies Headers (6) Tests Status: 200 OK Time: 28 ms

Pretty Raw Preview JSON

```
1 {
2   {
3     "customerID": "6",
4     "customerName": "Dian",
5     "customerCity": "semarang",
6     "customerPhone": "0810000000"
7   },
8   {
9     "customerID": "7",
10    "customerName": "Eko",
11    "customerCity": "Jogja",
12    "customerPhone": "0890000000"
13  },
14  {
15    "customerID": "8",
16    "customerName": "Rina",
17    "customerCity": "semarang",
18    "customerPhone": "0850000000"
19  },
20  {
21    "customerID": "13",
22    "customerName": "Joni",
23    "customerCity": "Jakarta",
24    "customerPhone": "08800000000"
25  }
26 }
```

Record baru hasil insert

Uji Coba API – DELETE (1)

Request → delete record

- URL: <http://localhost/server/index.php/api/customers>
- Method: DELETE
- Klik tab “Body”, pilih “x-www-form-urlencoded”, isikan parameter berikut:

- | Key | Value |
|-----|-------|
| id | 14 |

– Value sesuai dengan id record yg akan di delete

- Klik tombol “Send”

Uji Coba API – DELETE (2)

The screenshot displays a REST client interface with the following components:

- 2**: Method dropdown menu set to **DELETE**.
- 1**: URL input field containing `http://localhost/server/index.php/api/customers`.
- 6**: **Send** button to execute the request.
- 4**: Tab selector with **Body** highlighted.
- 3**: **Body** tab content area.
- 5**: Key-value table for the request body:

Key	Value
<input checked="" type="checkbox"/> id	14
New key	value
- 7**: **Body** tab content area showing the response JSON:

```
1 {  
2   "status": "succes"  
3 }
```
- Status: 201 Created**: Response status indicator.
- JSON**: Response format dropdown menu.

Hasil request DELETE → hapus data di server

Uji Coba API – PUT (1)

Request → update record

- URL: <http://localhost/server/index.php/api/customers>
- Method: PUT
- Klik tab “Body”, pilih “x-www-form-urlencoded”, isikan parameter berikut:

Key	Value
name	Rina
city	Semarang
phone	085
id	15

Field yang akan diupdate

- Klik tombol “Send”

Uji Coba API – PUT (2)

2 PUT 1 http://localhost/server/index.php/api/customers Params Send

Authorization 4 (1) Body 3 Pre-request Script Tests 6

☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary

Key	Value
<input checked="" type="checkbox"/> name	Rina
<input checked="" type="checkbox"/> phone	085
<input checked="" type="checkbox"/> city	Semarang
<input checked="" type="checkbox"/> id	15
New key	value

Body Cookies (1) Headers (6) Tests Status: 201 Created

Pretty Raw Preview JSON

```
1 {  
2   "status": "success"  
3 }
```

7 Hasil request PUT → update data di server